

Section 1: UAL Toolkit

□ Unified Accelerator Libraries

- Architecture
- Element-Algorithm-Probe Framework
- API interface

□ Interactive Analysis Extension

- Architecture
- Accelerator Physics Player
- Collection of Accelerator-specific viewers

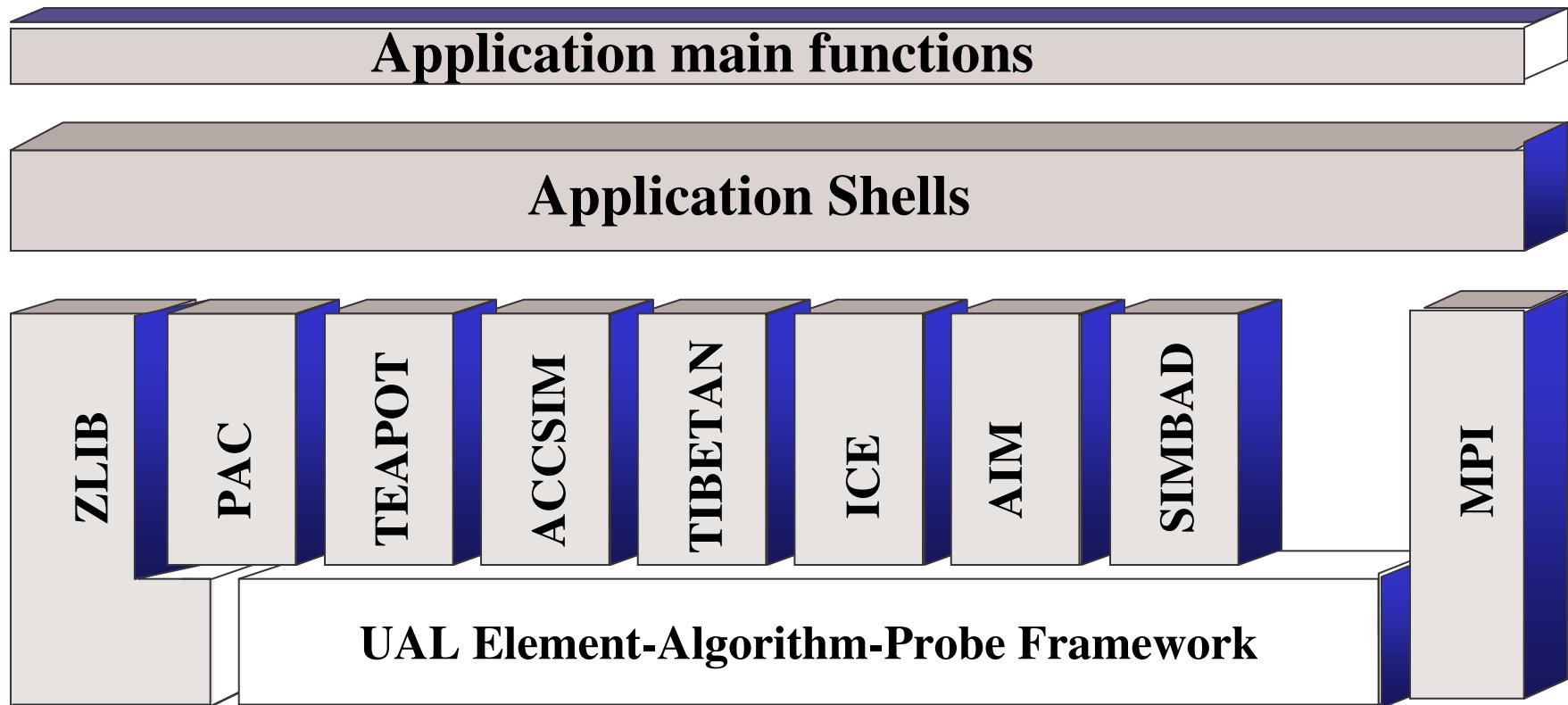
□ Directory Structure

UAL Objectives

ICAP 1996

- Form a customizable and extendable environment adaptable to new accelerator applications and conceptual models
- Facilitate development, deployment and reuse of diverse independently developed accelerator programs
- Integrate accelerator conceptual models and analysis patterns with modern technologies and software

UAL Environment



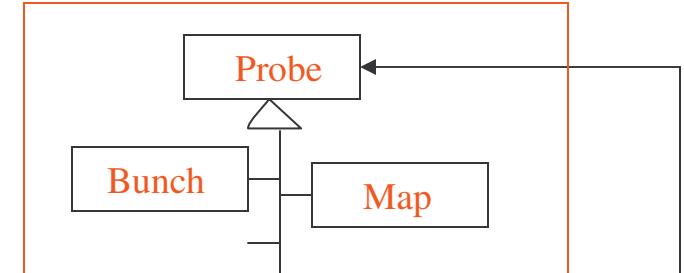
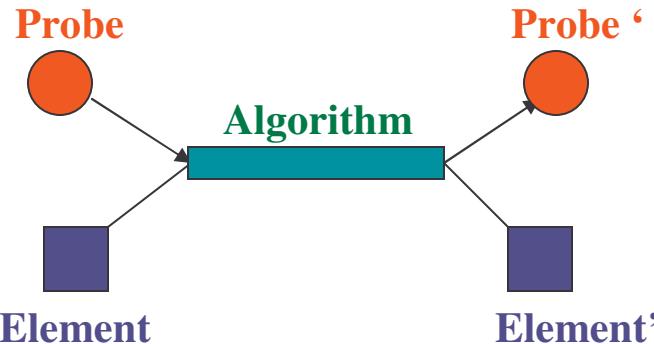
Libraries

<http://www.ual.bnl.gov>

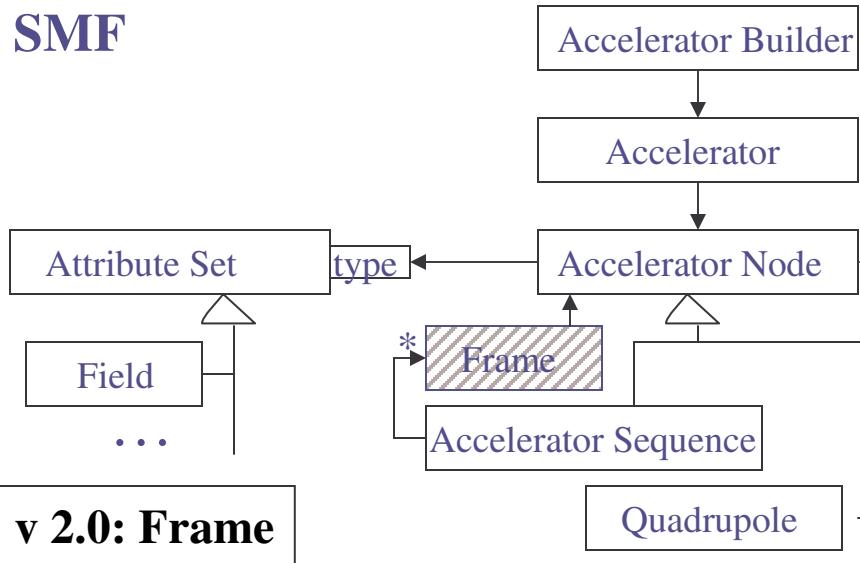
ACCSIM	<i>Accumulator Simulator (bunch distributions and diagnostics, collimator), author: Fred Jones, home page</i>
AIM	<i>Accelerator Instrumentation Models, authors: P.Cameron, M.Błaskiewicz</i>
BETACOOL	<i>Adapter to the BETACOOL code (collection of electron cooling, ibs, internal target algorithms, authors: Anatoly Sidorin, Alexander Smirnov, Grigori Trubnikov, home page)</i>
ICE	<i>Incoherent and Coherent Effects, author: M.Błaskiewicz</i>
PAC	<i>Platform for Accelerator Codes (collection of Common Accelerator Objects), authors: G.Bourianoff, N.Malitsky, A.Reshetov, R.Talman (SMF)</i>
SIMBAD	<i>Simulation of Beam Advanced Dynamics, authors: Alfredo Luccio and Nick D'Imperio,</i>
SPINK	<i>Tracking code for polarized particles in a circular accelerator, author: A.Luccio</i>
SXF	<i>The SXF represents a flat, complete, and independent description of the current accelerator state, authors: H.Grote, J.Holt, N.Malitsky, F.Pilat, R.Talman, C.G.Trahern, W.Fischer</i>
TEAPOT	<i>Thin Element Accelerator Program for Optics and Tracking (collection of symplectic trackers and mappers, collection of correction algorithms), authors: Richard Talman and Lindsay Schachinger</i>
TIBETAN	<i>A longitudinal phase space tracking program, author: J.Wei</i>
UAL	<i>UAL Element-Algorithm-Probe framework, authors: N.Malitsky, R.Talman</i>
UAL::QT	<i>QT-Based Interactive Analysis Environment of Unified Accelerator Libraries, authors: Valeri Fine, Nikolay Malitsky, Richard Talman</i>
UAL::ROOT	<i>Collection of ROOT-based (http://root.cern.ch) accelerator-specific viewers</i>
ZLIB	<i>Numerical library for differential algebra, author: Yiton Yan</i>

Element-Algorithm-Probe Framework

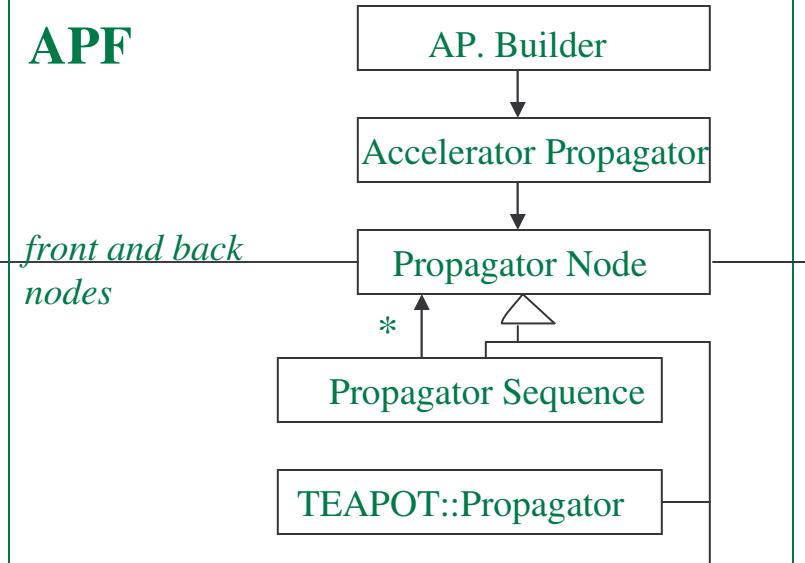
ICAP 1998



Standard Machine Format SMF



Accelerator Propagator Framework APF



Accelerator Description Exchange Format (ADXF 2.x)

<http://www.ual.bnl.gov/adxf/>

- ADXF file is a XML representation of the accelerator state ([The Element Concept](#))
- Its schema has been mapped from the SMF 2.x object model.



adxfsd

```
<xsd:schema ...>
  <xsd:include schemaLocation="adxfs_simple_types.xsd" />
  <xsd:include schemaLocation="adxfs_abstract_types.xsd" />
  <xsd:include schemaLocation="adxfs_core.xsd" />
  <xsd:include schemaLocation="adxfs_attribute_sets.xsd" />
  <xsd:include schemaLocation="adxfs_mad_elements.xsd" />
  <xsd:element name="adxfs">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="constants" minOccurs="0"
                     maxOccurs="unbounded" />
        <xsd:element ref="elements" minOccurs="0"
                     maxOccurs="unbounded" />
        <xsd:element ref="sectors" minOccurs="0"
                     maxOccurs="unbounded" />
      </xsd:sequence>
    ...
  </xsd:element>
</xsd:schema>
```

rhic.adxf

```
<adxfs .... >
  <constants>
    <constant name="lb" value="3.58896" />
    <constant name="ab" value="0.0151186" />
    ...
  </constants>
  <elements>
    <sbend name="d0mp08" l="lb" angle="ab" />
    <element name="bi8-dh0" design="d0mp08">
      <mfield b="0 0 0.005476 0.033503"
              a="0. 0 -0.010166 0.024366" />
    </element>
    ...
  </elements>
  <sectors>
    <sector name=blue">
      <frame ref="bi8-dh0" at="661.74662424" />
    ...
  </sector>
  </sectors>
</adxfs>
```

Position (local or global) of the installed component
which could be a common point for connecting with
the Detector description

Accelerator Propagator Description Format (APDF)

- APDF file is a XML representation of the accelerator propagator (The Algorithm Concept)
- Its schema is under development.

Applications range from small special tasks to full-scale realistic beam dynamics studies

Simple matrix-based tracker 

```
simple.apdf
<apdf>
  <propagator name="simple" accelerator="blue">
    <link algorithm="TIBETAN::SectorTracker"
          sector = "Default" />
    <link algorithm="TIBETAN::RFCavityTracker"
          elements="rfac1" />
    <link algorithm="TIBETAB::WCMonitor"
          elements="mend" />
  </propagator>
</apdf>
```

Element-by-Element tracker for Model Independent Analysis studies 

```
mia.apdf
<apdf>
  <propagator name="mia" accelerator="blue">
    <link algorithm="TEAPOT::DriftTracker"
          types = "Default" />
    <link algorithm="TEAPOT::DriftTracker"
          types="Marker|Drift|[VH]monitor|Monitor" />
    <link algorithm="TEAPOT::DipoleTracker"
          types="SBend" />
    <link algorithm="TEAPOT::MltTracker"
          types="Quadrupole|Sextupole|Multipole|Kicker" />
    <link algorithm="TIBETAN::RFCavityTracker"
          types="RfCavity" />
    <link algorithm="AIM::Monitor"
          types="Monitor" />
    ....
  </propagator>
</apdf>
```

User-oriented API

ROOT 2004 Workshop

```
// Declare UI Interface
UAL::QT::PlayerShell shell;

// ****
cout << "Read the ADXF file (lattice description)." << endl;
// ****
shell.readADXF(Args() << Arg("file", latticeFile ));

...
// ****
cout << "Read the APDF file (propagator description)." << endl;
// ****

shell.readAPDF(Args() << Arg("file", apdfFile ));

// ****
cout << "Generate a bunch distribution." << endl;
// ****

shell.setBunch(Args()
    << Arg("np", 10000)
    << Arg("enx", 15.0e-6)
    << Arg("eny", 15.0e-6)
    << Arg("ctMax", 2.0)
    << Arg("deMax", 1.0e-3)
    << Arg("seed", -100));
```

Evolution:

1994: C++ API

1996: Perl API

1998: Perl API + SXF lattice file

2003: Perl API + SXF lattice file
+ XML-based propagator file

2004: C++ API + SXF lattice file
+ XML-based propagator file

2005: C++ API + XML-based
lattice and propagator files

C++ with CINT vs Perl/Python:

Pros: debugging, maintenance

Cons: none

UAL Interactive Analysis Extension

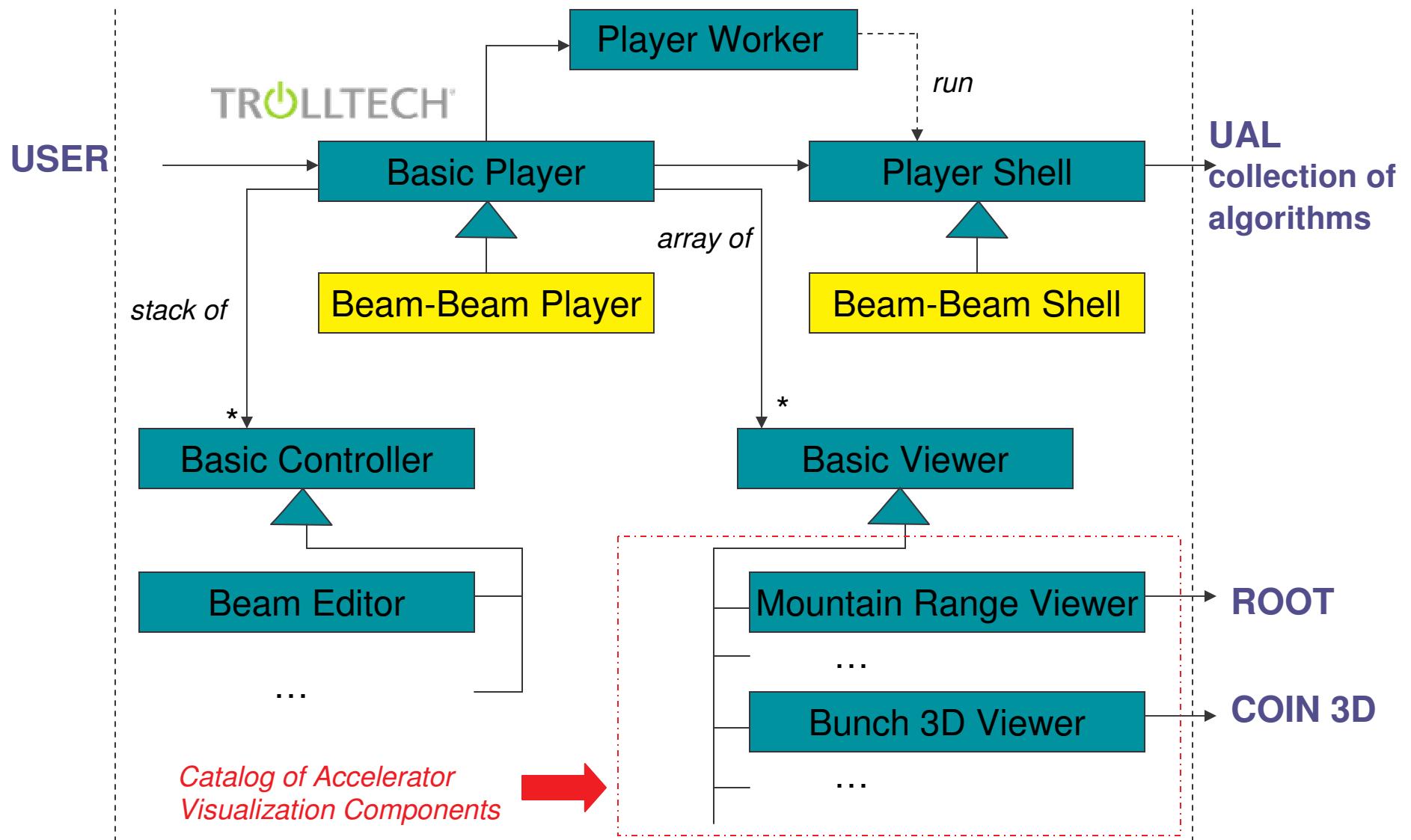
Objectives

- Bring the UAL off-line applications to the RHIC online environment for analyzing accelerator physics experiments and operational data.

- Facilitate modeling and analysis of multi-particle applications, such as beam-beam and space charge effects, instabilities, cooling, *etc.*)

Interactive Analysis Extension

Architecture



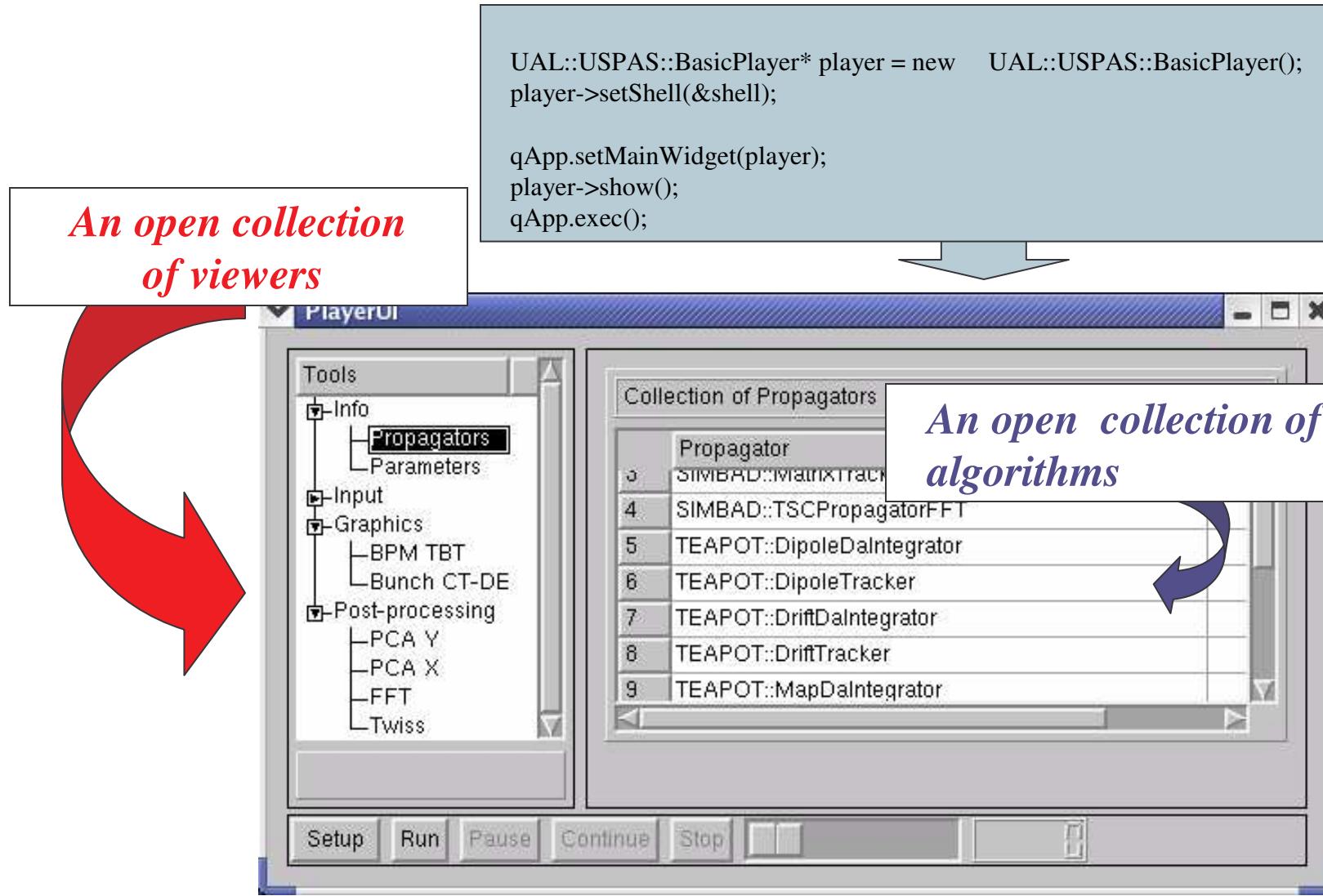
Interactive Analysis Extension (cont)

Architecture

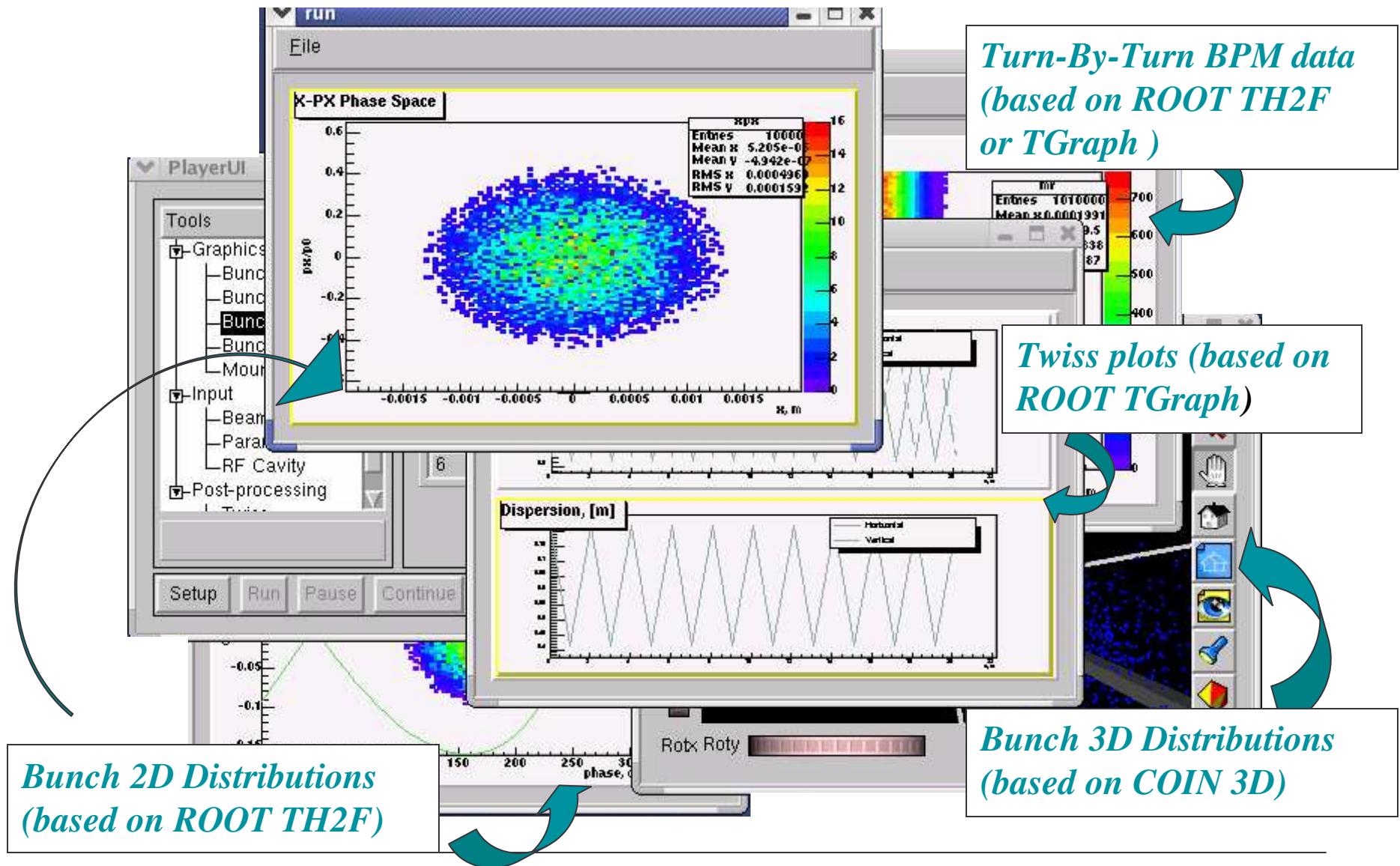
- **Player Shell** – non-GUI interface/façade to the UAL classes
- **Basic Player** – main widget of the Interactive Analysis Extension
- **Player Worker** – non-GUI thread performing time-consuming analysis and simulation
- **Basic Controller** – base class of the various editors for changing the UAL parameters
- **Basic Viewer** – base class of the visualization components

Accelerator Physics Player

ACAT 2005, May 22-27



(Some) Examples of the Accelerator-Specific Viewers



USPAS Directory Structure

Disk 1: modeling software

- **cshrc-tentative:** c-shell resource file to be copied to `~/.cshrc`
- **rootrc:** ROOT resource file to be copied to `~/.rootrc`
- **tools:** QT GUI framework (<http://www.trolltech.com>) and ROOT data analysis framework (<http://root.cern.ch>)
- **ual1:** Unified Accelerator Libraries (<http://www.ual.bnl.gov>)
- **examples:** USPAS examples

Disk 2: accelerator lattices and XML editors

- **oxygen5:** XML-aware editor, licensed for USPAS school
- **grace:** graphical post-processing tool
- **xslt:** collection of accelerator lattices in various formats (ADX, MAD8, SXF, *etc.*)

UAL1:

- **Makefile**
- **env**: Unix configuration files with environmental variables
- **tools**: common freely available libraries used by the UAL applications (e.g. Xerces, FastMathParser, FFTW, SXF, *etc.*)
- **codes**: unified accelerator physics libraries
- **ext**: extensions, such as User-oriented shell, ADXF and SXF parsers
- **gui**: Qt-based Accelerator Physics Player and ROOT/COIN3D visualization components
- **doc**: Doxygen-based specification of the UAL classes, ADXF schema
- **examples**: (incomplete) list of examples illustrating various features of the UAL environment

Examples:

- **Makefile**
- **lattices:** ADXF files, such as eq_tune_fodo.adxf, general_fodo.adxf, *etc.*
- **player:** Player used by all USPAS applications
- **transverse:** application of the chapters 3-4
- **longitudinal:** application of the chapter 6
- **decoherence:** application of the chapter 7
- **nonlinear:** application of the chapter 8